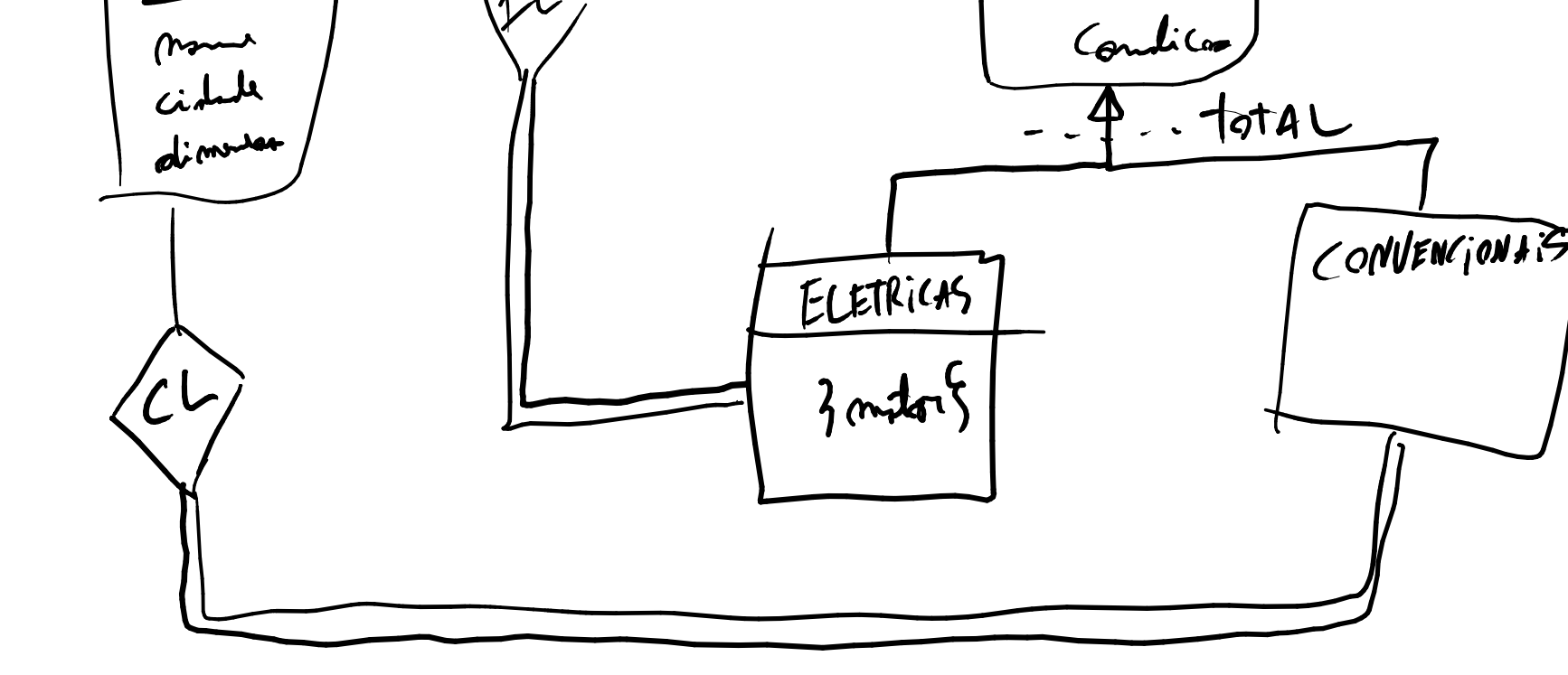


- I - V
- 2 - V
- 3 - F
- 4 - F
- 5 - V
- 6 - V
- 7 - F, não avaliado por unknown
- 8 - V
- 9 - V, a data finalmente só pode estar ordenada de um forma
- 10 - F

II
 Num B+tree as mäs estas ordenadas portanto o algoritmo ideal para fazer a procura é um binary search.
 Portanto em cada nó aplicamos um binary search para encontrar o índice do próximo nó ou o elemento que pretendemos encontrar.
 Neste caso em concreto, queremos procurar o nome 'Einstein'. Como alfabeticamente 'Einstein' < 'Mozart' procuramos no nó à esquerda de 'Mozart'.
 Nesse nó aplicamos o binary search para encontrar 'Einstein'.

III
 Superchaves é um conjunto de atributos que identifica inequivocamente qualquer instância de uma relação.
 Chave candidata é uma superchave mínima isto é, com o menor número de atributos.
 Chave primária é a chave candidata escolhida.
 Num relação podem existir várias superchaves.
 Num relação podem existir várias chaves candidatas.
 Cada relação tem apenas 1 chave primária.
 Exemplos: Professores (id, cc, nome)
 id é um identificador que é a chave primária da relação.
 cc é uma chave candidata da relação.
 O conjunto id, nome e cc é uma superchave da relação.

IV
 As propriedades ACID são atomicidade, consistência, isolamento e durabilidade.
 Atomicidade refere-se ao facto de uma transação ser completamente executada ou não ser executada de todo.
 Consistência refere-se ao facto de uma transação "transportar" o base de dados de um estado consistente para outro estado consistente.
 Isolamento refere-se ao facto de as transações desconhecem-se umas às outras, isto é, são executadas de forma independente uma da outra. Transações concorrentes não têm acesso a estados intermédios inconsistentes de outras transações.
 Durabilidade refere-se ao facto de após uma transação ser committed as alterações são permanentes, mesmo em caso de falha de software, hardware ou outro.
 Estas propriedades garantem que as bases de dados são sempre consistentes e válidas mesmo em caso de falha de energia, falha de software ou hardware, entre outros.



RI: As bicicletas convencionais podem estar associadas a um ou mais lojas, desde que sejam da mesma cidade.

VI
 a) E2 tem um n que o identifica e um y. Um E2 só pode ter um E3 ou um E4 (disposto total). Um E4 tem uma lista de z. Um E5 tem um n que o identifica e um y. Um E1 tem um n que o identifica e um g. Há uma relação A1 entre E1 e E3 com um atributo z. A cada E4 está associado um E5.

b) $E2(x, y) \rightarrow E4(z, w)$
 $E3(x) \rightarrow E1(x, y)$
 $FK(n) \rightarrow E2$
 $E4(x, w) \rightarrow A1(x, y, z)$
 $FK(n) \rightarrow E2$
 $FK(w) \rightarrow E5$
 $NN(w)$
 $FK(y) \rightarrow E3$

RI: E2 e E3 ou E4 e especializa-se sempre (disposto total)

VII
 a) $\pi_{nome} (\sigma_{idade > 20} (alunos))$
 b) $\pi_{nome, nome-curso} (\sigma_{des-identificacao = 'Económico'} (alunos \times cursos))$
 c) $LEI \leq \sigma_{nome-curso = 'LEI'} \pi_{nome, idade} (\sigma_{alunos, nome-curso = 'LEI'} (P_{a1}(alunos) \times P_{a2}(LEI)))$
 $NAOVALIDOS \leq \pi_{nome, idade} (\sigma_{alunos, nome-curso = 'LEI'} (alunos))$
 $TODOS \leq \pi_{nome, idade} (alunos)$
 $\pi_{nome-curso} (count(nome-alunos) as nome-alunos, avg(idade) as idade-media)$

VIII
 a) select nome from alunos where idade > 20;
 b) select alunos.nome, alunos.nome-curso from alunos, cursos where alunos.nome-curso = cursos.nome-curso and cursos.des-identificacao = 'Económico';
 c) select nome, idade from alunos where idade > all (select idade from alunos where nome-curso = 'LEI');
 d) select nome-curso, count(nome-alunos) as nome-alunos, avg(idade) as idade-media from alunos group by nome-curso;
 e) select distinct c1.des-identificacao from cursos c1 where not exists (select 1 from cursos c2 where c2.des-identificacao = c1.des-identificacao and c2.grau not in (select c3.grau from cursos c3 where c3.des-identificacao = c1.des-identificacao))